



Программный комплекс Систэм Платформ

SePlatform.Data Server 2.1 Модуль OPC AE Server

Руководство администратора

Редакция
3. Предварительная

Соответствует версии ПО
2.1.2



© ООО «СИСТЭМ СОФТ», 2022-2024. Все права защищены.

Авторские права на данный документ принадлежат ООО «СИСТЭМ СОФТ». Копирование, перепечатка и публикация любой части или всего документа не допускается без письменного разрешения правообладателя.

Содержание

1. Назначение и принцип работы	4
2. Функциональные возможности	6
2.1. Режимы работы модуля	6
2.2. Постановка сигнала на обслуживание	6
2.3. Генерация события	6
2.4. Атрибуты уведомлений	12
2.5. Рассылка уведомлений о квитировании	12
2.6. Блокирование и подавление	13
3. Настройка модуля	14
4. Работа с сигналами	16
5. Агрегация событий	18
5.1. Настройка агрегации событий	18
5.2. Настройка агрегации событий по ветке	19
5.3. Настройка фильтров агрегатора	21
5.4. Работа агрегаторов со ссылками	23
6. Передача событий по OPC DA	26
7. Описание файла конфигурации сервера	30
8. Пример работы с модулем	34
9. Диагностика работы модуля	35
Список терминов и сокращений	38

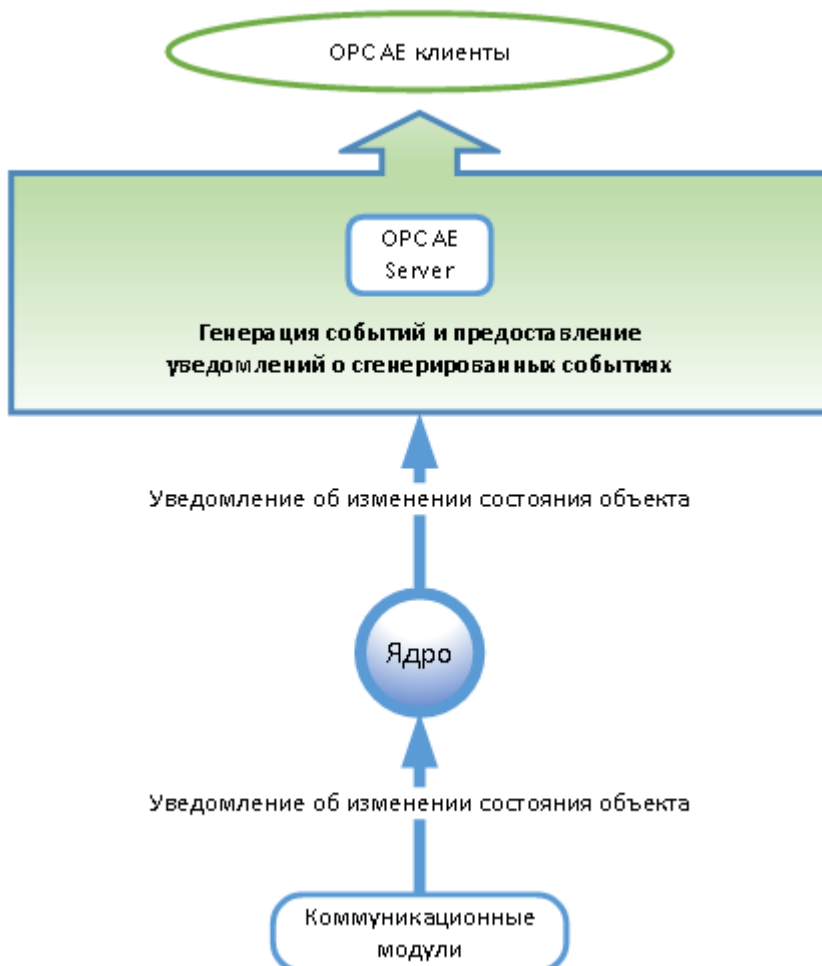
1. Назначение и принцип работы

Основные функции модуля OPC AE Server:

- генерация уведомлений о событиях по заданным условиям и передача сгенерированных уведомлений подключенным клиентам;
- прием информации о квитировании уведомлений о событиях от одного из клиентов и передача этой информации другим клиентам.

Работая в составе SePlatform.Data Server, модуль реализует серверную часть спецификации OPC AE. В составе SePlatform.Data Server может присутствовать только один модуль OPC AE Server.

Ядро получает от коммуникационного модуля уведомление об изменении состояния объекта и передает полученное уведомление модулю OPC AE Server, если у модуля есть подписка на пришедшее уведомление. Модуль OPC AE Server генерирует событие по пришедшему уведомлению, которое удовлетворяет условию генерации события. Генерация производится по типам условий ([стр. 6](#)). Источниками данных для генерации события являются сигналы SePlatform.Data Server. Клиент получает уведомление о сгенерированном событии от модуля OPC AE Server, в соответствии со спецификацией OPC AE.



Модуль OPC AE Server посылает уведомления клиенту, если:

- наступило новое событие, на которое подписан клиент;
- поступил запрос обновления списка активных условий от клиента. Уведомления о событиях придут клиенту с пометкой, что уведомления обновленные, а не оригинальные.

При старте модуля происходит инициализация модуля. Инициализация включает в себя:

- чтение конфигурационных данных модуля;
- подписка у ядра на получение уведомлений об изменениях сигналов;
- проверка корректности настройки адресов для сигналов и постановка их на обслуживание.

После инициализации модуль ожидает подключения клиента.

Модуль позволяет клиенту делить уведомления по категориям (**Discrete, Enumeration, Dynamic, Level, VSLevel, Deviation**), но области источника не различает. Другими словами, модуль не позволяет клиенту различать дерево сигналов источника.

2. Функциональные возможности

2.1. Режимы работы модуля

Модуль OPC AE Server может работать в двух режимах: **РАБОТА** или **РЕЗЕРВ**. Режим работы модуля в сервере устанавливает ядро. Режим работы модуля напрямую зависит от режима работы сервера. Режим работы сервера может измениться после резервных переходов.

В режиме **РАБОТА** модуль выполняет функции:

- получает уведомления об изменении состояния объекта;
- генерирует события;
- передает уведомления о событии клиенту;
- передает клиенту уведомление о квитировании.

Отличие режима **РЕЗЕРВ** от режима **РАБОТА** заключается в том, что модуль не передает уведомления о событиях клиенту, т.к. не происходит генерация событий.

При переходе из режима **РЕЗЕРВ** в режим **РАБОТА** модуль отправляет клиенту уведомления о событиях, пришедшие со времени работы модуля в режиме **РАБОТА**.

2.2. Постановка сигнала на обслуживание

Для постановки сигнала на обслуживание модулем OPC AE Server при запуске модуля выполняются следующие действия:

- проверка правильности задания адреса сигнала;
- проверка разрешений алармов для типа объекта;
- проверка наличия подусловий и разрешений подусловий.

2.3. Генерация события

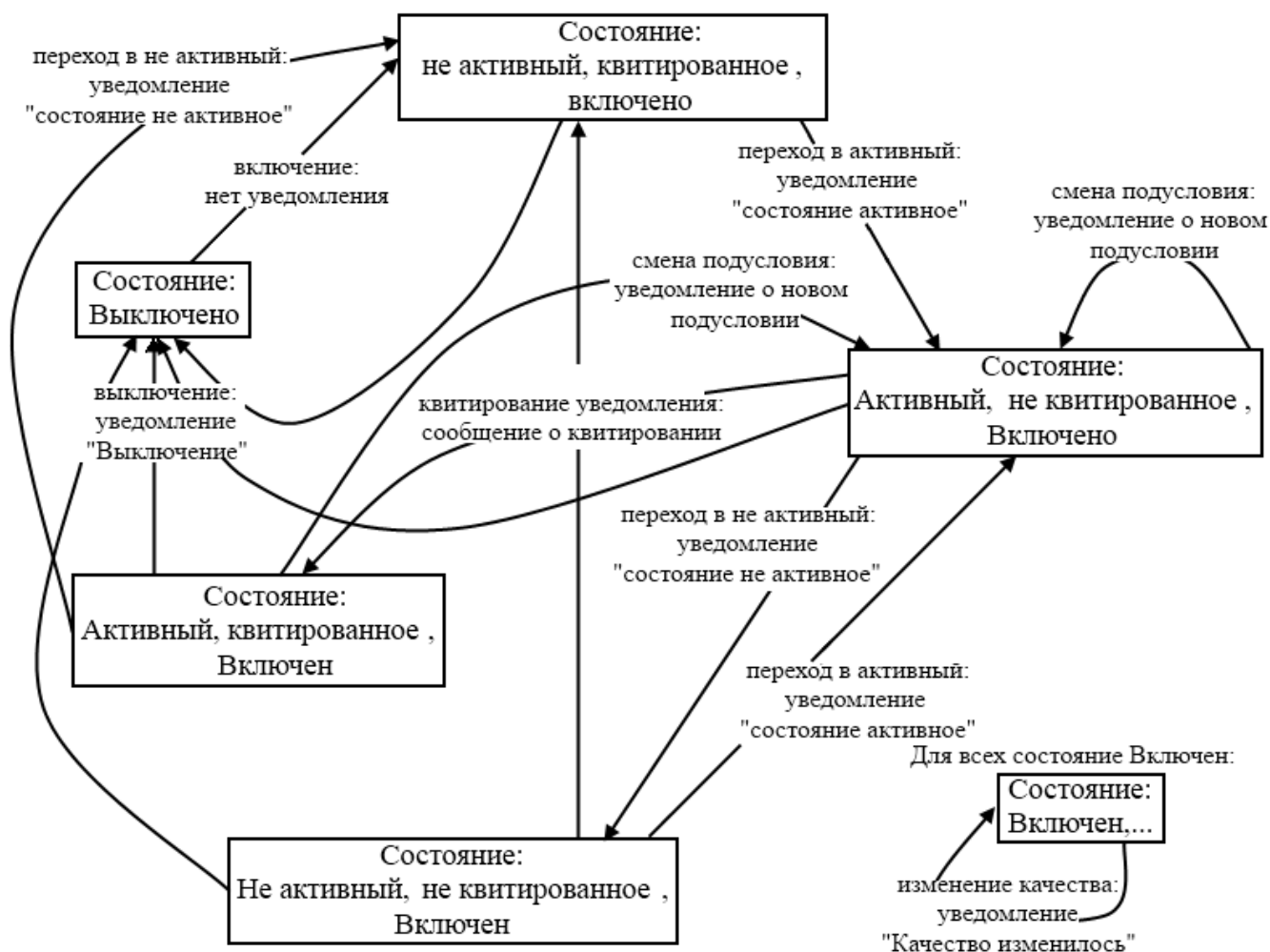
На рисунке ниже представлен алгоритм перехода между состояниями условий в виде графа.

Первоначальным состоянием условия при запуске модуля является состояние: **неактивное условие**, **квитированное**, **включено** для создания уведомлений.

Переход условия в состояние **активное** происходит при получении модулем уведомления о событии. При этом происходит проверка выполнения, какого либо из подусловий. Определив, какое из подусловий выполняется, модуль генерирует событие и посылает уведомление о событии клиенту.

Переход условия в состояние **включено** происходит при изменении в файле конфигурации ([стр. 30](#)) значения атрибута **Enabled** для тега **EventCondition**.

Переход условия в состояние **квитированное** происходит при подтверждении прочтения уведомления о событии клиентом ([стр. 12](#)).



Для модуля OPC AE Server источниками данных являются сигналы SePlatform.Data Server. С каждым источником данных может быть связано несколько подусловий. При выполнении любого подусловия OPC AE клиент получает уведомление о событии.

Модуль OPC AE Server реализует генерацию событий по следующим типам условий:

- перечисления;
- дискретные;
- динамические;
- события по уровню;
- события по произвольному количеству уровней;
- предельное отклонение.

Перечисления

События по перечислению генерируются в соответствии с идентификатором события, получаемым от стороннего источника данных (под сторонним источником подразумеваются любые источники данных сервера, которые реализованы вне модуля событий – любые коммуникационные модули, модуль вычислений и т.п.). Идентификатор содержит код подусловия, которое активируется и по которому формируется событие. Применяются при работе с численными типами объектов: uint1, int1, uint2, int2, uint4, int4, uint8, int8, float, double.

**ПРИМЕР**

Если получено значение «3», то выдать сообщение - **Значение в норме**, со звуковым сигналом из файла on.wav. Если получено значение «2», выдать сообщение - **Значение не пришло**, без звукового сигнала. Если получено значение «1», выдать сообщение - **Значение не верно**, со звуковым сигналом из файла err.wav.

**ОБРАТИТЕ ВНИМАНИЕ**

При настройке звука события необходимо в ссылке на звуковой файл указывать расширение *.wav.

Код условия **Enumeration**:

```
<EventConditions>
  <EventCondition Name="EnumCondition" Type="Enumeration" Enabled="1">
    <Subcondition Type="Enumeration" Value="3" Message="Значение в норме"
    Sound="on.wav" Severity="9" Enabled="1" SoundEnabled="1" />
    <Subcondition Type="Enumeration" Value="2" Message="Значение не пришло" Sound=""
    Severity="9" Enabled="1" SoundEnabled="0" />
    <Subcondition Type="Enumeration" Value="1" Message="Значение не верно"
    Sound="err.wav" Severity="9" Enabled="1" SoundEnabled="1" />
  </EventCondition>
</EventConditions>
```

Дискретные

События по дискретным параметрам содержат два независимых подусловия: **On** и **Off**. Для параметра они могут быть настроены, а могут отсутствовать. Подусловия активируются при соответствующем изменении значения параметра. Применяются при работе с типами объектов bool.

**ПРИМЕР**

Если состояние объекта изменилось с «True» на «False», выдать сообщение - **Отключение объекта**, со звуковым сигналом. Если состояние объекта изменилось с «False» на «True», выдать сообщение - **Включение объекта**, со звуковым сигналом.

Код условия **Discrete**:

```
<EventConditions>
  <EventCondition Name="DiscreteCondition" Type="Discrete" Enabled="1">
    <Subcondition Type="On" Message="Включение объекта" Sound="on.wav" Severity="9"
    Enabled="1" SoundEnabled="1" />
    <Subcondition Type="Off" Message="Отключение объекта" Sound="off.wav"
    Severity="1" Enabled="1" SoundEnabled="1" />
  </EventCondition>
</EventConditions>
```

Динамические

События по динамическим условиям, созданным пользователем. Подусловия активируются при соответствующем изменении значения параметра. Возможность настраивать события для динамических

сигналов сервера. Например, для сигналов, характеризующих состояние резервирования. Применяются при работе с типами объектов string.



ПРИМЕР

Если произошла смена пользователя, то выдать сообщение **Смена диспетчера**.

Код условия **Dynamic**:

```
<EventConditions>
  <EventCondition Name="DynamicCondition" Type="Dynamic" Enabled="1">
    <Subcondition Type="Dynamic" Message="Смена диспетчера" Sound="snd.wav"
Severity="100" Enabled="1" SoundEnabled="1" />
  </EventCondition>
</EventConditions>
```

Модуль OPC AE Server имеет возможность указать источники данных для генерации событий со стороны OPC DA клиента. Такими источниками выступают строковые сигналы типа string. В настройках модуля следует связать сигнал с типом условия, по которому будут генерироваться события.

Чтобы сгенерировать динамические алармы в тип данных string (строковый параметр) следует внести запись в значение XML конструкции вида:

```
<Subcondition Type="Dynamic" Message="DynMessage" Sound="snd.wav" Severity="100" Enabled="1"
SoundEnabled="1" />
```

События по уровню

События по уровням, созданным пользователем. Подусловия активируются при соответствующем изменении значения параметра.

Чтобы указать пределы, используются типы подусловий - **Normal**, **Hi**, **Lo**, **HiHi**, **LoLo**, **HiHin**, **LoLon** (n в диапазоне от «2» до «5»):

- **Normal** - подусловие, которое выполняется, если не активно ни одно из других подусловий;
- **Hi**, **HiHi** и **HiHin** - подусловия, которые выполняются, если значение параметра больше или равно указанного значения;
- **Lo**, **LoLo** и **LoLon** - подусловия, которые выполняются, если значение параметра меньше или равно указанного значения.



ПРИМЕР

Значение параметра устройства должно находиться в двух пределах: с «0.5» до «1.5» и с «1.5» и выше. Это означает, что для генерации событий будут использоваться 2 типа подусловий - **Hi** и **HiHi**. Уведомление пользователю будет генерироваться, только если полученное значение параметра попадет в один из указанных пределов. Однако, если от устройства придет повторное значение, не выходящее за текущий предел, повторного уведомления клиентам не будет. Например, если полученное значение изменилось с «0.5» до «1», то уведомления клиент не получит.

Код условия **Level**:

```
<EventConditions>
  <EventCondition Name="Port_1" Type="Level" Enabled="1">
    <Subcondition Type="Hi" Message=": Порт 1. Несанкционированное отключение. Сигнал снят" Value="0.5" Sound="" Severity="9" Enabled="1" SoundEnabled="0" />
    <Subcondition Type="HiHi" Message=": Порт 1. Несанкционированное отключение. Сигнал установлен" Value="1.5" Sound="" Severity="3" Enabled="1" SoundEnabled="0" />
  </EventCondition>
</EventConditions>
```

События по произвольному количеству уровней

События по уровням, созданным пользователем. Пользователь может задать произвольное количество подусловий. Подусловия активируются при соответствующем изменении значения параметра.

Чтобы указать пределы, используются типы подусловий - **Above**, **Below** и **Normal**:

- **Above** - подусловие, которое выполняется, если значение параметра больше или равно указанного значения;
- **Below** - подусловие, которое выполняется, если значение параметра меньше или равно указанного значения;
- **Normal** - подусловие, которое выполняется, если не активно ни одно из подусловий типов **Above**, **Below**.

Каждое такое подусловие должно иметь уникальное имя в рамках условия. Имя подусловия задается произвольно пользователем в атрибуте **Name** элемента **Subcondition**. Ниже показаны примеры уникальных имен подусловий:

- **UpperEmergency** - уровень выше аварийного;
- **UpperLimit** - уровень выше предельного;
- **UpperNormative** - уровень выше нормативного;
- **Normal** - уровень в норме;
- **LowerLimit** - уровень ниже предельного;
- **LowerEmergency** - уровень ниже аварийного.

Применяются при работе с численными типами объектов: uint1, int1, uint2, int2, uint4, int4, uint8, int8, float, double.

**ПРИМЕР**

Значение параметра, который отвечает за уровень загрузки процессора, должен находиться в четырёх пределах. Все возможные значения параметров от устройства могут быть в диапазоне от «0» до «100»: «0» - нормальное значение, «100» - аварийное значение. Это означает, что для генерации событий будут использоваться 4 типа подусловий типа **Above**. Уведомление пользователю будет генерироваться, только если полученное значение параметра попадет в один из указанных пределов. Однако, если от устройства придет повторное значение, не выходящее за текущий предел, повторного уведомления клиентам не будет. Например, если полученное значение изменилось с «55» до «65», то уведомления клиент не получит.

Код условия **VSLevel**:

```
<EventConditions>
  <EventCondition Name="st-comm1.AverageCPULoad" Type="VSLevel" Enabled="1">
    <Subcondition Name="Normal" Type="Normal" Message=". Нормальная загрузка процессора" Value="" Sound="" Severity="150" Enabled="1" SoundEnabled="0" />
    <Subcondition Name="Above50" Type="Above" Message=". Высокая загрузка процессора" Value="50" Sound="" Severity="250" Enabled="1" SoundEnabled="0" />
    <Subcondition Name="Above70" Type="Above" Message=". Критическая загрузка процессора" Value="70" Sound="" Severity="500" Enabled="1" SoundEnabled="0" />
    <Subcondition Name="Above85" Type="Above" Message=". Аварийная загрузка процессора" Value="85" Sound="" Severity="800" Enabled="1" SoundEnabled="0" />
  </EventCondition>
</EventConditions>
```

Предельное отклонение

События с предельным отклонением содержат два независимых подусловия: **Up** и **Down**. Подусловия активируются при соответствующем изменении значения параметра. Применяются при работе с численными типами объектов: uint1, int1, uint2, int2, uint4, int4, uint8, int8, float, double.

**ПРИМЕР**

Если произошло резкое повышение текущего значения на «5», выдать сообщение - **Предельное отклонение вверх**. Если произошло резкое понижение текущего значения на «10», выдать сообщение - **Предельное отклонение вниз**.

**ПРИМЕЧАНИЕ**

Предельное отклонение считается от текущего значения, не от начального.

Код условия **Deviation**:

```
<EventConditions>
  <EventCondition Name="AEDeviation" Type="Deviation" Enabled="1">
    <Subcondition Type="Up" Message="Предельное отклонение вверх" Value="5" Sound="Количество датчиков.wav" Severity="667" Enabled="1" SoundEnabled="1" />
    <Subcondition Type="Down" Message="Предельное отклонение вниз" Value="10" Sound="" Severity="667" Enabled="1" SoundEnabled="1" />
  </EventCondition>
</EventConditions>
```

2.4. Атрибуты уведомлений

Модуль OPC AE Server предоставляет клиентам возможность просматривать уведомления с дополнительной информацией. Дополнительная информация представляет собой атрибуты сообщения.

Отправленные модулем уведомления для клиента включают следующие атрибуты:

- **Source** - источник события - полное имя тега, от которого произошло событие;
- **Time** - время, когда сгенерировано уведомление о событии;
- **EventCategory** - категория события - коды категорий событий:
 - «0» - **Discrete**;
 - «1» - **Enumeration**;
 - «2» - **Dynamic**;
 - «3» - **Level**;
 - «3» - **VSLevel**;
 - «4» - **Deviation**.
- **Type** - тип события - существуют следующие типы событий:
 - **Discrete**;
 - **Enumeration**;
 - **Dynamic**;
 - **Level**;
 - **VSLevel**;
 - **Deviation**.
- **Severity** - приоритет - важность данного события;
- **Message** - сообщение - текст, который описывает событие.

Отправленные модулем уведомления для клиента помимо стандартных атрибутов имеют следующие дополнительные атрибуты, которые отображаются в OPC AE клиенте:

- **ConditionName** - имя условия генерации события;
- **SubConditionName** - имя подусловия генерации события;
- **ChangeMask** - маска изменений - признак показывает, какие условия изменились, чтобы вызвать сервер для отправки уведомления о событии;
- **NewState** - индикатор нового состояния условия показывает новые значения свойств условий: разрешений, активности, квитирования;
- **ConditionQuality** - качество условия показывает качество основных элементов данных, на которых это условие основывается;
- **AckRequired** - требование квитирования - флаг, определяющий требуется ли квитирование от клиента уведомления о событии;
- **ActiveTime** - время перехода состояния события в активное (время, когда произошло событие);
- **Cookie** - текстовая информация о событии, хранящаяся в сервере. Информация является недоступным для клиента;
- **ActorID** - идентификатор OPC клиента, который квитирует событие.

2.5. Рассылка уведомлений о квитировании

Модуль получает уведомление о квитировании события от клиента и отправляет его остальным клиентам, подписанным на это событие.



Уведомления о квитировании события содержат следующую информацию:

- имя пользователя;
- дата и время квитирования события;
- примечания пользователя.

2.6. Блокирование и подавление

Модуль OPC AE Server предоставляет клиентам возможности блокирования и подавления источников или объектов.

При блокировании источника или объекта модуль OPC AE Server:

- прекращает генерацию событий данного источника или объекта;
- прекращает сохранение в историю событий данного источника или объекта;
- прекращает уведомление клиентов о событиях данного источника или объекта.

Блокирование источников или объектов выполняется бессрочно, пока блокировка не будет отменена клиентом.

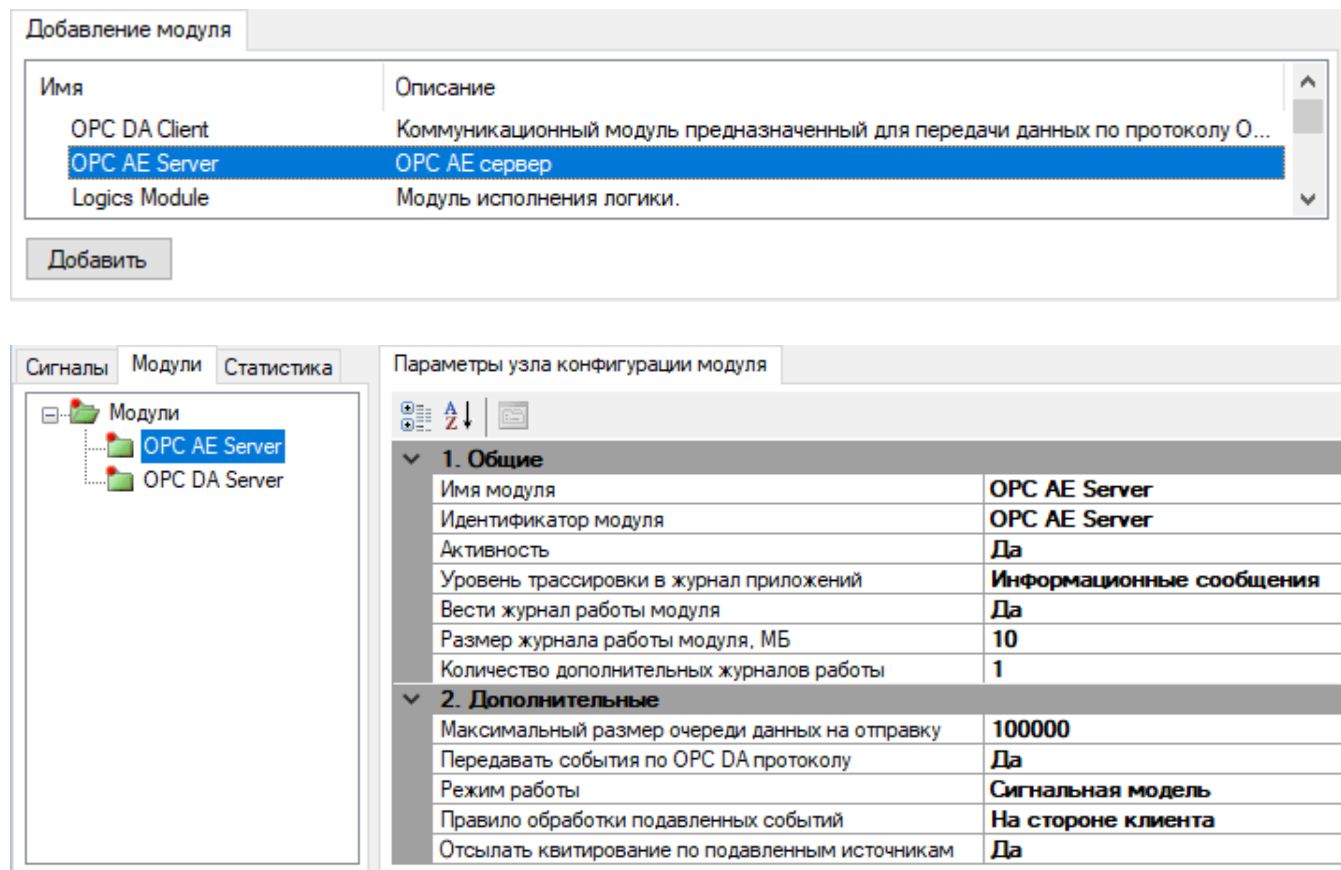
При подавлении источника или объекта модуль OPC AE Server:

- продолжает генерацию событий данного источника или объекта;
- продолжает сохранение в историю сгенерированных событий данного источника или объекта;
- прекращает уведомление клиентов о событиях данного источника или объекта.

Подавление источника или объекта возможно на любой период времени или бессрочно, пока подавление не будет отменено клиентом.

3. Настройка модуля


Чтобы добавить модуль в состав конфигурации сервера, используйте сервисное приложение Конфигуратор.



Общие параметры модуля OPC AE Server совпадают с общими параметрами всех модулей SePlatform.Data Server.

Дополнительные параметры модуля:

Параметр	Описание
Максимальный размер очереди данных на отправку	Количество уведомлений, стоящих в очереди на генерацию уведомлений о событии для клиентов. При достижении максимального количества уведомлений приостанавливается работа модуля. Модуль продолжает работу, если очередь освободилась хотя бы на одно уведомление. Значение параметра находится в диапазоне от «10 000» до «10 000 000». По умолчанию параметр принимает значение «100 000»
Передавать события по OPC DA протоколу	<p>Доступ к событиям и квитирование событий по протоколу OPC DA:</p> <ul style="list-style-type: none"> ➤ «Да» - доступ предоставляется; ➤ «Нет» - доступ не предоставляется. <p>Доступ предоставляется через массив динамических сигналов (стр. 26), с которыми можно работать через любой OPC DA клиент. Чтобы массив динамических сигналов отобразился в дереве сигналов OPC DA клиента, перезапустите SePlatform.Data Server.</p>

Параметр	Описание
Режим работы	<p>Режим работы модуля:</p> <ul style="list-style-type: none"> ➤ «Сигнальная модель» - выбирается, когда объекты автоматизации и их параметры выражаются в виде сигналов, для формирования дерева сигналов используется SePlatform.Data Server; ➤ «Объектная модель» - выбирается в случае применения объектно-ориентированного подхода к построению проекта с помощью SePlatform.Development Studio. <div>  <p>ПРИМЕЧАНИЕ Режим работы модуля зависит от способа построения проекта автоматизации.</p> </div>
Правило обработки подавленных событий	<p>Выбор стороны, на которой выполняется обработка подавленных событий:</p> <ul style="list-style-type: none"> ➤ «На стороне сервера» - модуль OPC AE Server не уведомляет клиентов о подавленных событиях; ➤ «На стороне клиента» - модуль OPC AE Server уведомляет клиентов о событиях, независимо от состояния подавления.
Отсылать квитирование по подавленным источникам	<p>Рассылка уведомлений о квитировании событий по подавленным источникам:</p> <ul style="list-style-type: none"> ➤ «Да» - уведомления о квитировании событий рассылаются клиентам даже если источники подавлены; ➤ «Нет» - уведомления о квитировании событий не рассылаются. <p>Значение по умолчанию «Нет».</p>

4. Работа с сигналами

Сигналы сервера используются для передачи значений параметров технологических объектов пользователю и для передачи управляющих воздействий пользователя к технологическим объектам. Сигналы привязываются к модулям через свойство адрес сигнала.

Сигнал состоит из набора свойств. Часть свойств описывает параметры технологического объекта (к ним относятся значение, качество, метка времени, а так же опциональные свойства - свойства пересчета), а другая часть описывает поведение сигнала в сервере (к ним относятся обязательное свойство **CDT**).

С точки зрения модуля необходимы следующие свойства сигнала:

- **1 (CDT)** - канонический тип данных;
- **2 (Value)** - инженерное значение;
- **5000 (Address)** - адрес сигнала;
- **999000 (ObjectType)** - тип объекта;
- **999001 (ObjectCode)** код объекта;
- **999002 (ObjectSound)** - звук объекта;
- **999003 (EventsEnabled)** - признак генерации сообщений;
- **999004 (Conditions)** - условия генерации сообщений;
- **999005 (IsAbstract)** Тип абстрактный или нет.

Помимо вышеперечисленного необходимые свойства объекта следующие:

- **1 (CDT)** - канонический тип данных;
- **2 (Value)** - инженерное значение;
- **3 (Quality)** - качество сигнала;
- **4 (Timestamp)** - метка времени.

Свойства сигналов

Свойства сигнала для модуля:

- **5000 (Address)** - адрес сигнала. Свойство создается пользователем. Сигнал может быть привязан к одному или нескольким модулям. Связь осуществляется через указание адреса в рамках спецификации протокола в свойстве **5000 (Address)**). Значение свойства адрес сигнала для каждого модуля заключается в фигурные скобки {}. Значение параметра помещается в круглые скобки, а между именем параметра и значением ставится знак =.

В одном свойстве может находиться несколько адресов, записанных в формате {адрес_1}{адрес_2}...{адрес_N}.

Структура формата адреса для модуля OPC AE Server представлена в таблице:

ModuleId	Идентификатор модуля	Строка уникальная в пределах сервера. По умолчанию: « OPC AE Server »
Protocol	Протокол/спецификация	« OPC AE »
Conditions	Список условий	Список имен условий объекта, разделенных запятыми. Имена условий должны совпадать со значением атрибута Name для тега EventCondition в файле конфигурации (<Имя сервера>ExternalObjects.xml)



ПРИМЕР

Адрес сигнала.

```
{ModuleId=(OPC AE Server) Protocol=(OPCAE) Conditions=(Status damper) }
```

При пересечении или совпадении адресов (например, при подключении одинакового условия генерации события по одному объекту) происходит запись в журнал модуля о наличии пересечений адресов, но к отказу в постановке сигналов на обслуживание не приводит. Проверка на совпадение и пересечение адресов происходит при старте модуля.

- **999000 (ObjectType)** - имя объекта. Свойство создается пользователем. Указывает, что сигнал или папка являются объектом, с которого поступает информация о событии. Свойство используется для связки сигнала, для которого необходимо генерировать сообщения о состоянии с условием генерации сообщения в конфигурации. Тип поля строковый (string). Значение параметра должно совпадать со значением атрибута **Name** для тега **Type** в файле конфигурации ([стр. 30](#)) (SePlatform.ServerExternalObjects.xml);
- **999001 (ObjectCode)** - код объекта;
- **999002 (ObjectSound)** - звуковой файл объекта. Каждый раз при срабатывании события для этого объекта перед звуком самого события будет добавляться звук объекта. В качестве значения используется строка-имя файла. Строка задается в формате: <имя файла>.wav. Файл должен располагаться в директории C:\ProgramData\System Platform\Alarms\Sounds и обязательно иметь формат *.wav. При получении уведомления о событии если заданы звук объекта и звук события, то первым воспроизведется звук объекта, а после звук события;
- **999003 (EventsEnabled)** - признак доступности генерации события по подусловию. Вычисляется по всей структуре генерации событий;
- **999004 (Conditions)** - условия генерации сообщений о событии. Содержит имя условия генерации сообщения о событии. Должно совпадать со значением атрибута **Name** для тега **EventCondition** в файле конфигурации ([стр. 30](#)) (SePlatform.ServerExternalObjects.xml);
- **999005 (IsAbstract)** - признак указывающий является ли данный объект абстрактным или не является. Если параметр принимает значение «1», то тип объекта абстрактный, если параметр принимает значение «0» - не абстрактный. Экземпляры абстрактного типа создавать нельзя.

Настройка сигналов

Для настройки сигналов модуля OPC AE Server воспользуйтесь сервисным приложением Конфигуратор.

Для добавления сигнала:

1. На вкладке **Сигналы** создайте сигнал нужного типа.
2. На вкладке **Свойства** сигнала нажмите **Добавить**.
3. В окне **Добавление свойства** выберите из выпадающих списков **Номер** и **Тип** сигнала.
4. В поле **Значение** укажите значение свойства сигнала ([стр. 16](#)).
5. Перезапустите службу **SePlatform.Server**.

5. Агрегация событий

Агрегация событий - объединение событий определенного объекта в единую информационную структуру.

Ниже приведено несколько примеров использования агрегации событий на мнемосхеме:

- сигнализирование о наличии неквитированных событий в объекте;
- квитирование сразу всех активных событий объекта;
- индикация в зависимости от максимальной важности среди активных событий объекта;
- индикация, если не все события объекта были квитированы.

5.1. Настройка агрегации событий

Агрегация событий происходит по родительской области событий. Также возможно настроить агрегацию для событий по нескольким родительским областям ([стр. 19](#)).

Чтобы настроить агрегацию событиям, необходимо:

- добавить в состав SePlatform.Data Server модуль OPC AE Server и настроить его;
- добавить агрегатор событий для объекта генерации.

Добавление и настройка модуля OPC AE Server

В Конфигураторе выполните следующие действия:

1. Добавьте в состав конфигурации и активируйте модуль OPC AE Server.
2. В параметрах узла конфигурации модуля выберите режим работы **Сигнальная модель**.

Добавление агрегатора событий для объекта генерации

1. Внутри объекта генерации событий создайте папку для агрегатора. Название папки может быть произвольным, для наглядности такая папка будет называться **Aggregator**.

The screenshot displays the 'Свойства сигнала' (Signal Properties) window. The left pane shows a hierarchical tree of signals. The 'Aggregator' folder is selected under the 'ZDV1' object. The right pane shows a table with the following data:

Номер	Имя	Значение	Описание
8002			
999000	Object Type		Тип объекта

At the bottom of the right pane, there are three buttons: 'Добавить' (Add), 'Изменить' (Edit), and 'Удалить' (Delete).

2. Агрегатору задайте свойства:

- **8002** (тип string) - наличие данного свойства определяет, что данный объект представляет агрегатор событий родительской области событий. В данное свойство помещается описание фильтров агрегатора ([стр. 21](#)). В случае, если для агрегатора не требуется настройка фильтров, свойство остаётся пустым;
- **999000 (ObjectType)** - наличие данного свойства определяет, что данная папка является объектом.

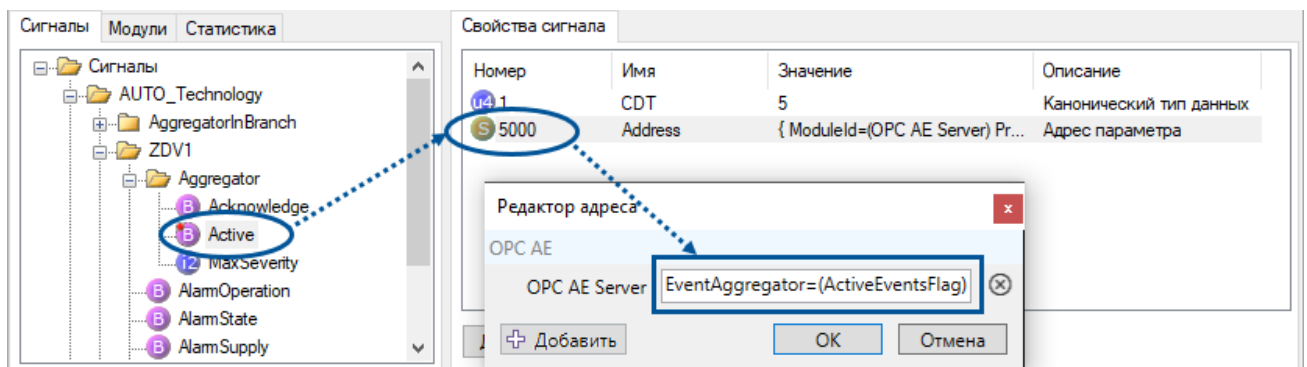
3. Дочерним узлам объекта агрегатора опишите параметры агрегации. Они представляют собой узлы определённого типа, которым посредством адреса сигнала (свойство **5000**) указывается тип агрегации:

```
EventAggregator=({тип параметра агрегатора})
```

Существуют следующие типы параметров агрегации объекта:

- **ActiveEventsFlag** (тип bool) - флаг наличия активных событий. Параметр принимает значение «true», если область содержит хотя бы одно активное событие;
- **MaxSeverity** (тип int2) - максимальная важность среди активных событий объекта. Если активных событий нет, параметр равен «0»;
- **Acknowledge** (тип bool) - при изменении значения на «true» отдаётся команда квитирования всех событий объекта. Дочерние сигналы **User** и **Comment** определяют пользователя и комментарий квитирования;
- **AllAcknowledged** (тип bool) - индикатор квитирования всех событий по текущему объекту. Если значение параметра равно «true», то в объекте квитированы все события. Если значение «false», то объект имеет неквитированные события.

На рисунке ниже показан пример описания параметра агрегации **ActiveEventsFlag**.

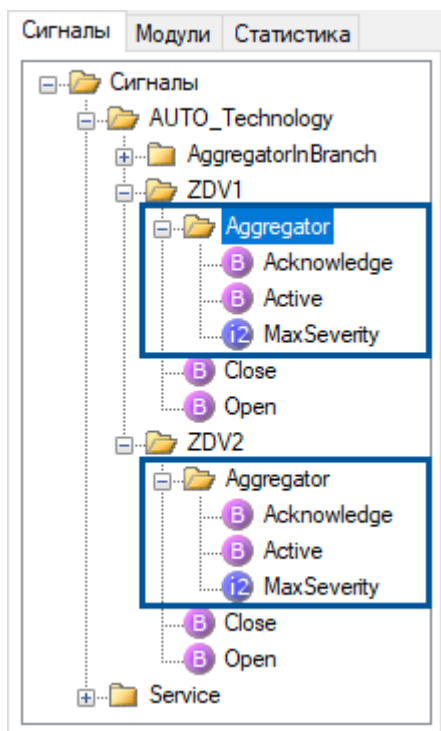


5.2. Настройка агрегации событий по ветке

Чтобы настроить агрегацию событиям по ветке, свяжите дочерние объекты агрегации с агрегатором.

Чтобы настроить агрегацию событий по ветке:

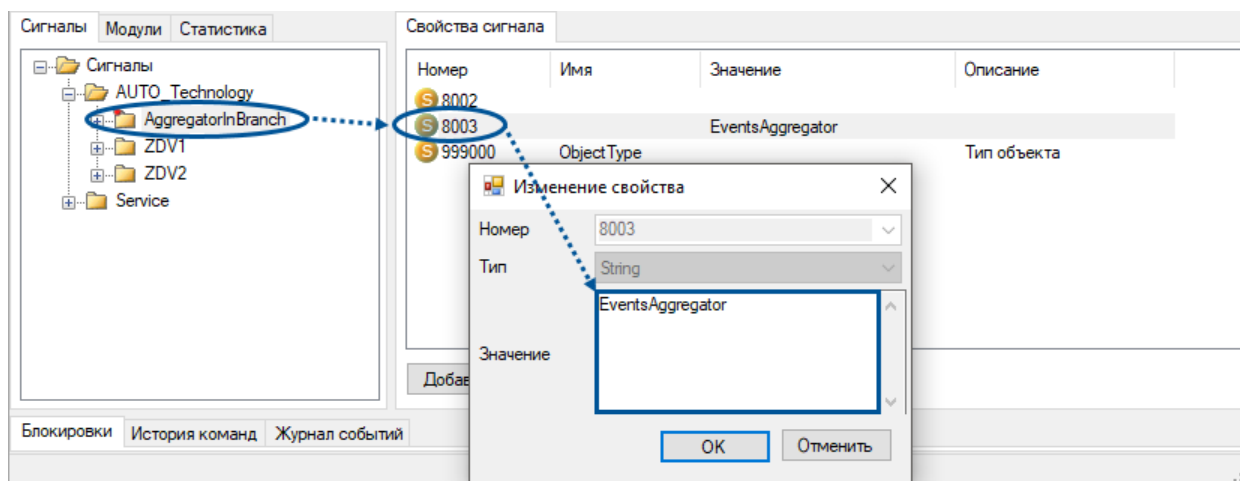
1. Добавьте и настройте агрегаторы событий для двух объектов ([стр. 18](#)). Например, для объектов **ZDV1** и **ZDV2** настройте агрегаторы квитирования всех активных событий, наличия активных событий и максимальной важности среди активных событий.



2. Создайте папку для агрегатора по ветке. Название папки может быть произвольным, для наглядности такая папка будет называться AggregatorInBranch.

3. Созданному агрегатору по ветке задайте свойства:

- **8002** (тип string) - наличие свойства определяет, что данный объект представляет агрегатор событий родительской области событий;
- **8003** (тип string) - данное свойство используется для связывания отдельных агрегаторов. В нем указывается идентификатор типа (имя произвольное, например, EventsAggregator);



- **999000** (Object Type) - наличие данного свойства определяет, что данная папка является объектом.

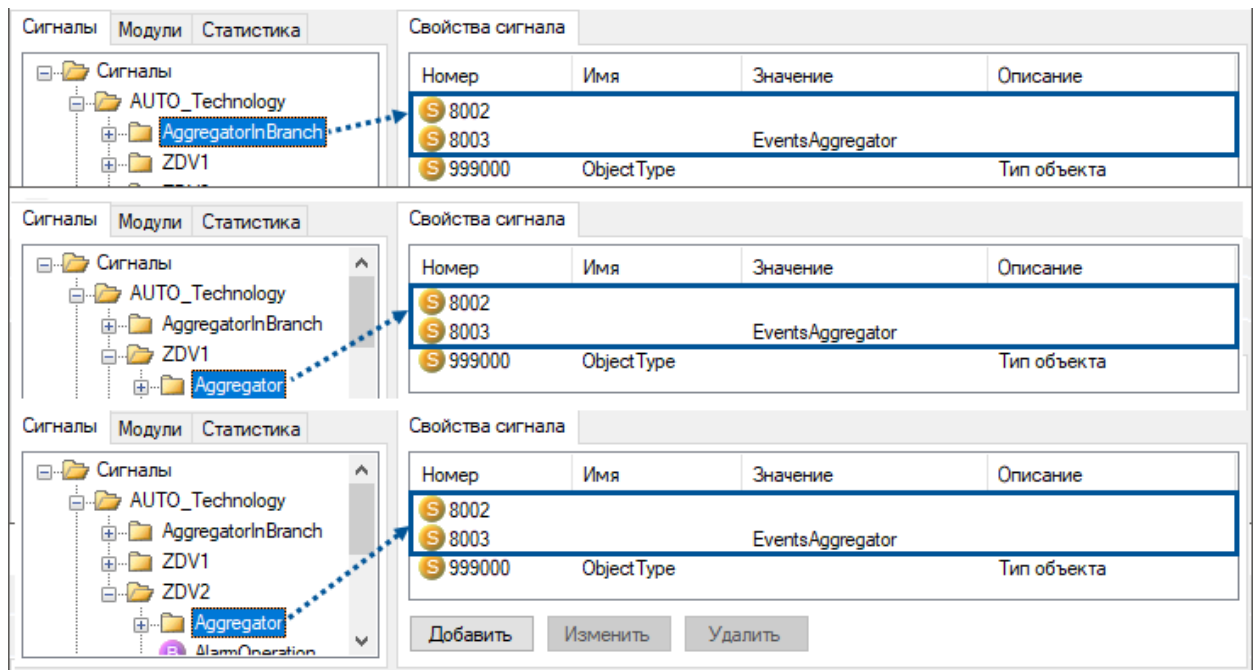
4. Для дочерних узлов объекта агрегатора по ветке опишите параметры агрегации ([стр. 18](#)).

Существуют следующие типы параметров агрегации объекта для агрегации по ветке:

- **ActiveEventsFlagInBranch** (тип bool) - флаг наличия активных событий в ветке;
- **MaxSeverityInBranch** (тип int2) - максимальная важность среди активных событий в ветке;
- **AcknowledgeInBranch** (тип bool) -команда для квитирования всех активных событий в ветке;
- **AllAcknowledgedInBranch** (тип bool) - индикатор квитирования всех событий в ветке. Если значение параметра равно «true», то все события ветки квитированы. Если значение «false», то есть неквитированные события в ветке;
- **ActiveObjects** (тип string) - список идентификаторов активных объектов. При изменении дочернего сигнала **Build** запускается сбор идентификаторов из всех активных объектов ветки. Результат будет помещён в сигнал в формате JSON.

5. Во всех объектах ветки, по которым требуется агрегация, должны быть описаны агрегаторы:

- одинакового типа, т.е. значение свойства **8003** должно быть такое же, как у агрегатора по ветке;
- с дублированием или переопределением фильтров ([стр. 21](#)), т.е. значение свойства **8002** может быть такое же как у агрегатора по ветке или отличным от него.



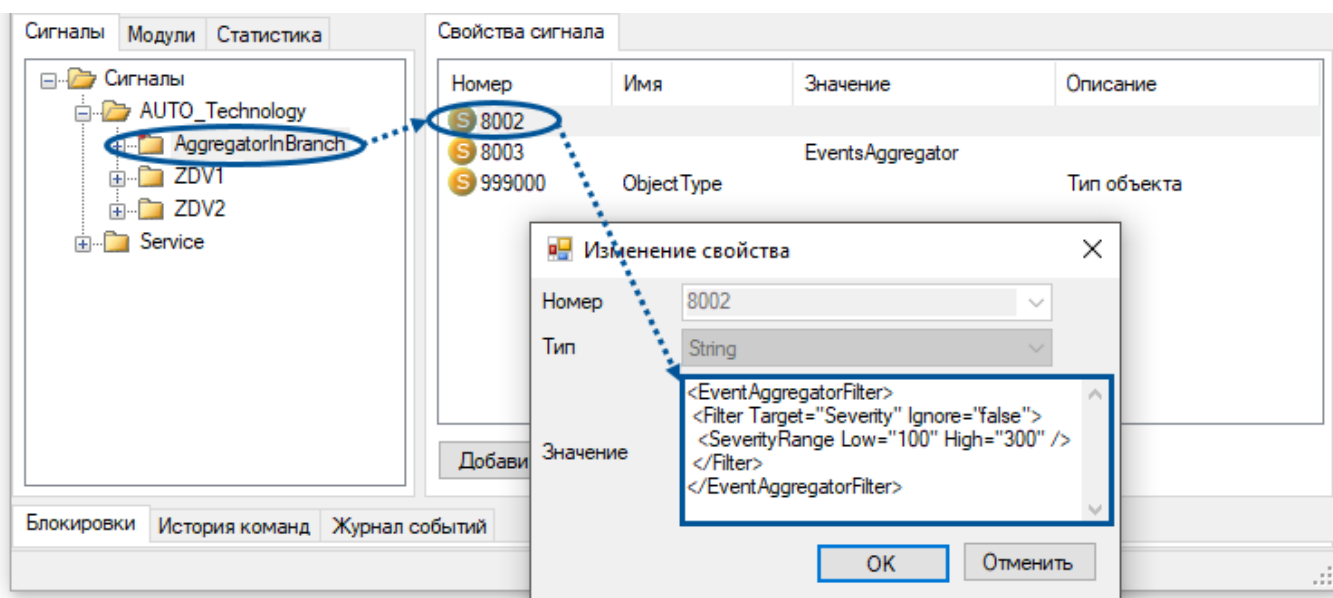
5.3. Настройка фильтров агрегатора

Для агрегатора можно настроить фильтры:

- по условиям генерации;
- по важности событий.

Фильтры агрегатора указываются в свойстве **8002** (тип string).

Вид фильтрации указывается в атрибуте **Target** (элемент **Filter**). Значение **Severity** - фильтр по важности событий, **Condition** - по условиям генерации. Чтобы фильтрация происходила по указанным условиям генерации или важности событий, укажите атрибуту **Ignore** значение «false». Чтобы фильтрация происходила по условиям генерации или важности событий кроме указанных, укажите атрибуту **Ignore** значение «true».



ПРИМЕР

Пример фильтрации по условиям генерации и по важности событий.

Условие фильтрации:

- важность события находится в диапазоне от «100» до «199» и от «200» до «299»;
- событие сгенерировано по любому типу условия, кроме типа **Dynamic** (динамический).

Код условия для фильтра агрегатора:

```
<EventAggregatorFilter>
<Filter Target="Severity" Ignore="false">
<SeverityRange Low="100" High="199" />
<SeverityRange Low="200" High="299" />
</Filter>
<Filter Target="Condition" Ignore="true">
<Condition Name="Dynamic">
<Subcondition Name="Сообщение"/>
</Condition>
</Filter>
</EventAggregatorFilter>
```



ПРИМЕР

Пример фильтрации по условиям генерации.

Условие фильтрация:

- важность события находится в диапазоне от «0» до «1000»;
- событие сгенерировано по условию типа Level (по уровню).

Код условия для фильтра агрегатора:

```
<EventAggregatorFilter>
<Filter Target="Condition" Ignore="false">
<Condition Name="Level">
<Subcondition Name="CISCO IE-3000-8TC"/>
</Condition>
</Filter>
</EventAggregatorFilter>
```



ПРИМЕР

Пример фильтрации по важности событий.

Условие фильтрация:

- важность события находится в диапазоне от «700» до «1000»;
- событие сгенерировано по любому типу условия.


Код условия для фильтра агрегатора:

```
<EventAggregatorFilter>
<Filter Target="Severity" Ignore="false">
<SeverityRange Low="700" High="1000" />
</Filter>
</EventAggregatorFilter>
```

5.4. Работа агрегаторов со ссылками

Агрегаторы могут работать со ссылками на объекты проекта. Чтобы настроить работу агрегаторов со ссылками, используйте следующие сигнальные свойства:

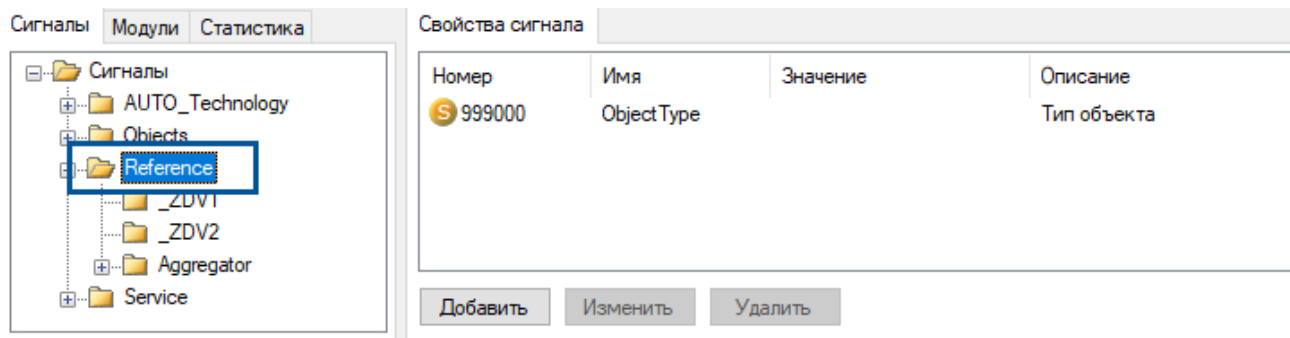
6001	string	полное имя объекта, к которому ведёт данная ссылка
6002	uint4	<p>Разновидность ссылки.</p> <p>Значения свойства:</p> <ul style="list-style-type: none"> ➤ 0 - ссылка ведёт исключительно на указанный объект и не затрагивает его поддерево сигналов; ➤ 1 - ссылка ведёт на указанный объект и его поддерево сигналов.

6003	bool	<p>Автораскрытие ссылки.</p> <p>Значения свойства:</p> <ul style="list-style-type: none"> ➤ «true» - разрешение на раскрытие поддерева объекта в DA-клиенте; ➤ «false» - запрещение на раскрытие поддерева объекта в DA-клиенте. 
6004	bool	<p>Константность ссылки. Значения свойств:</p> <ul style="list-style-type: none"> ➤ « true» - сигналы объекта, на который ведёт ссылка, доступны только для чтения в DA-клиенте; ➤ «false» - сигналы объекта, на который ведёт ссылка, доступны для изменений через DA-клиент.
6005	bool	<p>Если у объекта определено данное свойство (значение «true»), то агрегатор области, в которой определена ссылка, агрегирует также события целевого объекта</p>

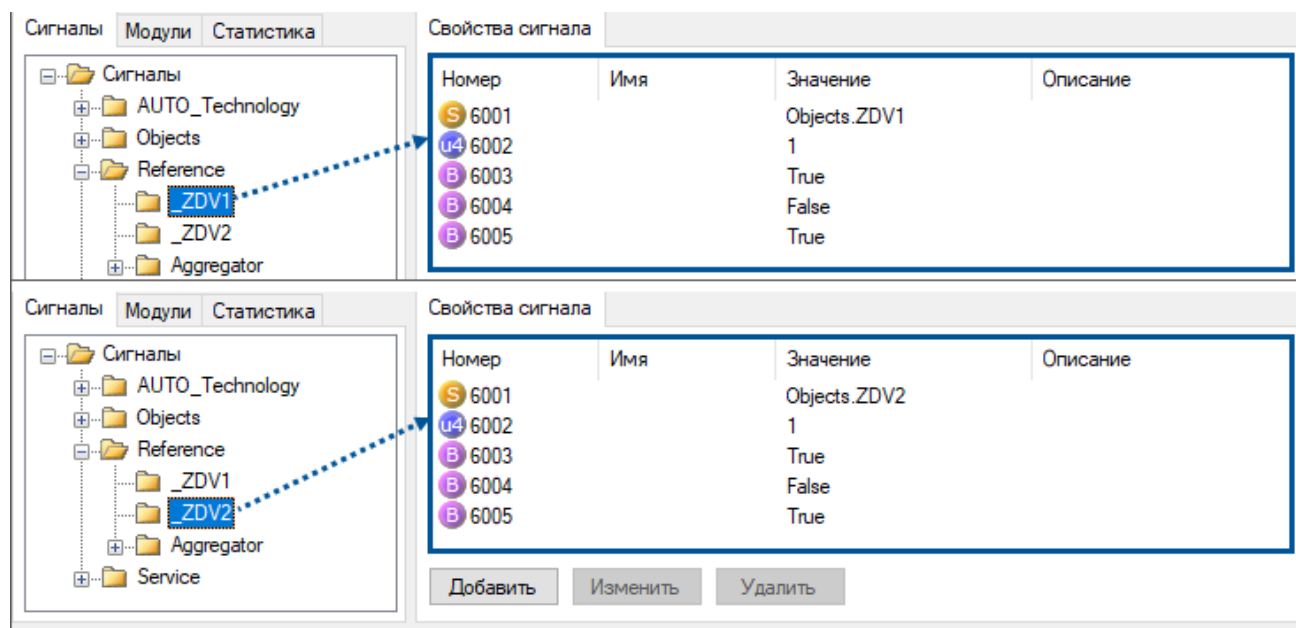
Пример работы агрегаторов со ссылками

В конфигурации проекта имеются несколько объектов, по которым необходимо настроить одинаковую генерацию событий. Чтобы не создавать каждому объекту свой агрегатор, будем использовать ссылки:

1. Создайте папку, внутри которой будут содержаться ссылки на объекты и агрегатор событий. Название папки может быть произвольным, для наглядности такая папка будет называться **Reference**.



2. В созданную папку добавьте папки, которые будут содержать в себе ссылки на объекты проекта. Объекты проекта находятся в папке **Objects**.

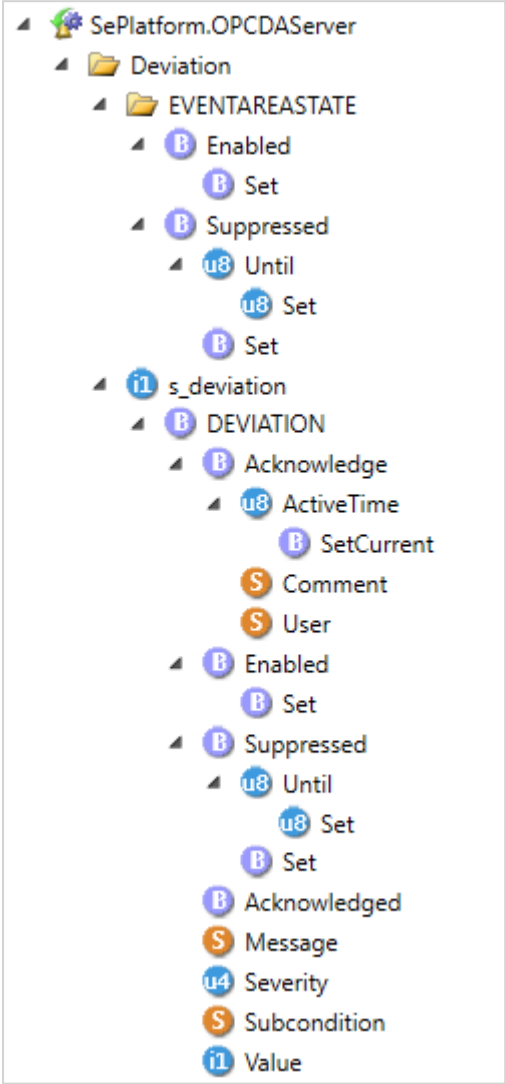


3. В папки **_ZDV1** и **_ZDV2** добавьте следующие свойства:

- **6001** (string). В значение свойства укажите полный путь до объекта, на который будет указывать ссылка;
- **6002** (uint4). В значение свойства укажите «1», чтобы ссылка указывала как на сам объект, так и на дочерние его объекты;
- **6003** (bool). В значение свойства укажите «true», чтобы с помощью DA-клиента иметь возможность раскрыть поддерево объекта;
- **6004** (bool). В значение свойства укажите «false», чтобы с помощью DA-клиента иметь возможность изменять значения сигналов;
- **6005** (bool). В значение свойства укажите «true», чтобы агрегатор событий начал агрегировать события объекта, на которые указывает ссылка.


6. Передача событий по OPC DA

Реализация мнемосхем предполагает предоставление информации о событиях по протоколу OPC DA. Если для сигнала определена генерация событий, а в настройках модуля OPC AE Server параметру **Передавать события по OPC DA протоколу** установлено значение «Да», то у сигнала появляется поддерево динамических сигналов, отражающих информацию о событиях.



Динамические сигналы:

Имя (Тип)	Описание
Сигналы подавления и блокирования событий объекта (папка «EVENTAREASTATE»)	
«Enabled» (bool)	Состояние блокировки событий объекта: <ul style="list-style-type: none">➤ «True» - события не блокированы;➤ «False» - события блокированы.

Имя (Тип)	Описание
<ul style="list-style-type: none"> » «Set» (bool) 	<p>Команда блокировки событий объекта:</p> <ul style="list-style-type: none"> » «True» - снять блокировку; » «False» - заблокировать.
«Suppressed» (bool)	<p>Состояние подавления событий объекта:</p> <ul style="list-style-type: none"> » «True» - события подавлены; » «False» - события не подавлены.
<ul style="list-style-type: none"> » «Set» (bool) 	<p>Команда подавления событий объекта:</p> <ul style="list-style-type: none"> » «True» - подавить; » «False» - снять подавление.
<ul style="list-style-type: none"> » «Until» (uint8) 	Метка времени в UTC планируемого снятия подавления объекта в формате FILETIME.
<ul style="list-style-type: none"> » «Set» (uint8) 	<p>Установка метки времени в UTC планируемого снятия подавления объекта.</p> <p>Для активации подавления до указанной метки времени сигналу «Suppressed.Set» объекта необходимо установить значение «True».</p> <p>Значение «0» - бессрочное подавление.</p>
Динамические сигналы источника событий	
«DEVIATION» (bool) (имя сигнала соответствует названию типа условия)	<p>Активность условия:</p> <ul style="list-style-type: none"> » «True» - условие активно; » «False» - условие неактивно.
«Acknowledge» (bool)	<p>Команда квитирования:</p> <ul style="list-style-type: none"> » «True» - квитировать событие с параметрами, заданными в сигналах «ActiveTime», «Comment», «User». <div>  <p>ОБРАТИТЕ ВНИМАНИЕ</p> <p>Квитирование выполняется только если сигналам «ActiveTime» и «User» установлено хорошее качество и значение.</p> </div>
<ul style="list-style-type: none"> » «ActiveTime» (uint8) 	Время активации условия.
<ul style="list-style-type: none"> » «SetCurrent» (bool) 	<p>Команда установки времени активации условия:</p> <ul style="list-style-type: none"> » «True» - в сигнал ActiveTime записывается время последней активации условия.
<ul style="list-style-type: none"> » «Comment» (string) 	Комментарий квитирования.
<ul style="list-style-type: none"> » «User» (string) 	Пользователь, который выполнил квитирование.

Имя (Тип)	Описание
«Enabled» (bool)	Состояние блокировки событий сигнала: <ul style="list-style-type: none"> ➤ «True» - события не блокированы; ➤ «False» - события блокированы.
➤ «Set» (bool)	Команда блокировки событий сигнала: <ul style="list-style-type: none"> ➤ «True» - снять блокировку; ➤ «False» - заблокировать.
«Suppressed» (bool)	Состояние подавления событий сигнала: <ul style="list-style-type: none"> ➤ «True» - события подавлены; ➤ «False» - события не подавлены.
➤ «Set» (bool)	Команда подавления событий сигнала: <ul style="list-style-type: none"> ➤ «True» - подавить; ➤ «False» - снять подавление.
➤ «Until» (uint8)	Метка времени в UTC планируемого снятия подавления сигнала в формате FILETIME.
➤ «Set» (uint8)	Установка метки времени в UTC планируемого снятия подавления сигнал. Для активации подавления до указанной метки времени сигналу «Supressed.Set» необходимо установить значение «True». Значение «0» - бессрочное подавление.
«Acknowledged» (bool)	Флаг квитирования: <ul style="list-style-type: none"> ➤ «True» - событие квитировано; ➤ «False» - событие не квитировано.
«Message» (string)	Сообщение события.
«Severity» (uint4)	Важность события.
«Subcondition» (string)	Имя активного подусловия.
«Value» (тип совпадает с типом сигнала источника события)	Значение сигнала.

Для работы с массивом динамических сигналов подключитесь к SePlatform.Data Server любым OPC DA клиентом.

Обозреватель

Поиск

SePlatform.OPCDA Server

Deviation

EVENTAREASTATE

s_deviation

DEVIATION

Acknowledge

ActiveTime

SetCurrent

Comment

User

Enabled

Set

Suppressed

Until

Set

Set

Acknowledged

Message

Severity

Subcondition

Value

s_deviation1

s_deviation2

s_deviation3

s_deviation4

s_deviation5

Инспектор 1

Тип	Сигнал	Значение
B	Deviation.EVENTAREASTATE.Enabled	True
B	Deviation.EVENTAREASTATE.Suppressed	False
B	Deviation.EVENTAREASTATE.Enabled.Set	False
u8	Deviation.EVENTAREASTATE.Suppressed.Until	0
B	Deviation.EVENTAREASTATE.Suppressed.Set	False
u8	Deviation.EVENTAREASTATE.Suppressed.Until.Set	0
i1	Deviation.s_deviation	
B	Deviation.s_deviation.DEVIATION	False
B	Deviation.s_deviation.DEVIATION.Acknowledge	False
B	Deviation.s_deviation.DEVIATION.Enabled	True
B	Deviation.s_deviation.DEVIATION.Suppressed	False
B	Deviation.s_deviation.DEVIATION.Acknowledged	False
S	Deviation.s_deviation.DEVIATION.Message	
u4	Deviation.s_deviation.DEVIATION.Severity	0
S	Deviation.s_deviation.DEVIATION.Subcondition	
i1	Deviation.s_deviation.DEVIATION.Value	
u8	Deviation.s_deviation.DEVIATION.Acknowledge.ActiveTime	0
S	Deviation.s_deviation.DEVIATION.Acknowledge.Comment	
S	Deviation.s_deviation.DEVIATION.Acknowledge.User	
B	Deviation.s_deviation.DEVIATION.Acknowledge.ActiveTime.SetCurrent	False
B	Deviation.s_deviation.DEVIATION.Enabled.Set	False
u8	Deviation.s_deviation.DEVIATION.Suppressed.Until	0
B	Deviation.s_deviation.DEVIATION.Suppressed.Set	False
u8	Deviation.s_deviation.DEVIATION.Suppressed.Until.Set	0

Всего 25 сигналов

7. Описание файла конфигурации сервера

Файл конфигурации сервера написан на языке XML, при заполнении файла следует придерживаться синтаксиса языка. Имя файла строится следующим образом: <Имя сервера>ExternalObjects.xml.



ОБРАТИТЕ ВНИМАНИЕ

После изменения файла конфигурации перезапустите сервер.

Теги, используемые при создании файла:

- **ExternalObjects** - корневой тег, имеющий атрибуты:
 - **CreateDate** - дата создания файла;
 - **UserName** - имя пользователя, создавшего файл;
- **Types** - данные, содержащиеся в теге, представляют собой список всех типов объектов;
- **Type** - тег для объявления объекта.

Атрибуты:

- **Name** - имя объекта, для которого необходимо получать сообщения о событии;
- **IsAbstract** - признак абстракции объекта;
- **EventsEnabled** - флаг, предназначенный для включения/отключения сообщений по данному объекту;
- **EventConditions** - данные, содержащиеся в теге, представляют собой список условий;
- **EventCondition** - условия генерации сообщений о событии.

Атрибуты:

- **Name** - имя условия для генерации сообщений о событии;
- **Type** - описание типа условия. Может принимать следующие значения:
 - **discrete** указывает на дискретный тип данных (bool);
 - **enumeration** указывает на численный тип данных (uint1, int1, uint2, int2, uint4, int4, uint8, int8, float, double);
 - **dynamic** указывает на динамический тип данных, созданный для пользовательских нужд (string);
 - **level** указывает на события по уровням (uint1, int1, uint2, int2, uint4, int4, uint8, int8, float, double);
 - **vslevel** указывает на события по уровням (uint1, int1, uint2, int2, uint4, int4, uint8, int8, float, double);
 - **deviation** указывает на события предельных значений (uint1, int1, uint2, int2, uint4, int4, uint8, int8, float, double);
- **Enabled** - флаг, предназначенный для включения/отключения условия генерации сообщения;

➤ **Subcondition** - подусловие генерации сообщений о событии.

Атрибуты:

- **Type** - описание типа условия генерации. Может принимать следующие значения:
 - для **Discrete** - «On» или «Off»;
 - для **Enumeration** - «Enumeration»;
 - для **Dynamic** - «Dynamic»;
 - для **Level** - «Normal», «Hi», «Lo», «HiHi», «LoLo», «HiHin», «LoLon» (n в диапазоне от «2» до «5»);
 - для **VSLevel** - «Above», «Below», «Normal»;
 - для **Deviation** - «Up» или «Down».
- **Message** - сообщение, выдаваемое при выполнении подусловия. Сообщение может содержать символы латиницы и кириллицы, а также цифры и необходимые знаки;
- **Value** - значение сигнала, необходимое для выполнения подусловия;
- **Sound** - звуковой файл, исполняемый при выполнении подусловия. Файл должен находиться в папке алармов и иметь расширение *.wav. Расширение указывать необязательно. Допускается указывать несколько звуковых файлов через запятую для последовательного воспроизведения;
- **SoundEnabled** - флаг, влияющий на значение атрибута **Sound**.

Если значение равно «1», к значению атрибута **Sound** добавляются звуки родительских объектов.



ПРИМЕР

Правила формирования значения атрибута **Sound**.

Конфигурация сервера содержит объекты с настроенными списками условий, подусловий и звуков:

- LU_21_24
 - KP_001
 - SW_1
 - SW_2

Если атрибут **SoundEnabled** в каком-либо подусловии для объекта **SW_1** равен «1», то значение атрибута **Sound** в этом подусловии равно:

Группа звуков объекта **LU_21_24** + Группа звуков объекта **KP_001** + Звук объекта **SW_1**.

Если значение равно «0», значение атрибута **Sound** очищается;

➤ **Severity** - приоритет сообщения. Возможные значения от «1» до «1000». Распределение уровня важности по диапазону значений приведено в таблице ниже. С увеличением значения уровень важности подусловия повышается;

Важность	Диапазон значений
Высокая	от 801 до 1000
Умеренно высокая	от 601 до 800
Средняя	от 401 до 600
Умеренно низкая	от 201 до 400
Низкая	от 1 до 200

- **AckRequired** - флаг, предназначенный для указания необходимости квитирования события. При отсутствии атрибута считается, что событие необходимо квитировать;
- **Enabled** - флаг, предназначенный для включения/отключения подусловия для генерации сообщений о событии;
- **Children** - список дочерних свойств. Содержание списка отсутствует;
- **SubTypes** - данные, содержащиеся в теге, представляют собой список подтипов объекта.

Пример фрагмента файла конфигурации сервера <Имя сервера>ExternalObjects.xml для модуля OPC AE Server:

```
<?xml version="1.0" encoding="windows-1251" standalone="no"?>
<!--Время создания: 20.09.2016 9:07:37, пользователь - Администратор, компьютер - -->
<ExternalObjects>
  <Types>
    <Type Name="User" IsAbstract="0" EventsEnabled="1">
      <EventConditions>
        <EventCondition Name="Alarm" Type="Dynamic" Enabled="1">
          <Subcondition Type="Dynamic" Message="" Value="" Sound=""Severity="100"
Enabled="1" SoundEnabled="1" />
        </EventCondition>
      </EventConditions>
    </Type>
    <Type Name="TU" IsAbstract="0" EventsEnabled="1">
      <EventConditions>
        <EventCondition Name="WaveAlarm" Type="Discrete" Enabled="1">
          <Subcondition Type="On" Message=" отклонения по смежным датчикам!" Value=""
Sound="" Severity="700" Enabled="1" SoundEnabled="1" />
        </EventCondition>
        <EventCondition Name="OnLineAlarmN" Type="Discrete" Enabled="1">
          <Subcondition Type="On" Message=" уменьшение количества датчиков на связи!"
Value="" Sound="" Severity="500" AckRequired ="0" Enabled="1" SoundEnabled="1" />
        </EventCondition>
        <EventCondition Name="OnLineAlarm0" Type="Discrete" Enabled="1">
          <Subcondition Type="On" Message=" нет датчиков на связи!" Value="" Sound=""
Severity="600" Enabled="1" SoundEnabled="1" />
        </EventCondition>
        <EventCondition Name="ControlAlarm" Type="Dynamic" Enabled="1">
          <Subcondition Type="Dynamic" Message=" поставлен на контроль" Value="" Sound=""
Severity="400" Enabled="1" SoundEnabled="1" />
        </EventCondition>
      </EventConditions>
    </Type>
    <Children />
    <SubTypes />
    <Type Name="P" IsAbstract="0" EventsEnabled="1">
      <EventConditions>
        <EventCondition Name="NaporHi" Type="Discrete" Enabled="1">
          <Subcondition Type="On" Message=" выше допустимого значения!" Value="" Sound=""
Severity="500" AckRequired ="1" Enabled="1" SoundEnabled="1" />
        </EventCondition>
      </EventConditions>
    </Type>
  </Types>
</ExternalObjects>
```



```
<EventCondition Name="StatusHi" Type="Discrete" Enabled="1">
  <Subcondition Type="On" Message=" выше допустимого значения!" Value="" Sound=""
Severity="500" AckRequired ="1" Enabled="1" SoundEnabled="1" />
</EventCondition>
<EventCondition Name="StatusLo" Type="Discrete" Enabled="1">
  <Subcondition Type="On" Message=" ниже допустимого значения!" Value="" Sound=""
Severity="500" Enabled="1" SoundEnabled="1" />
</EventCondition>
</EventConditions>
<Children />
<SubTypes />
</Type>
</Types>
<Children />
</ExternalObjects>
```

8. Пример работы с модулем

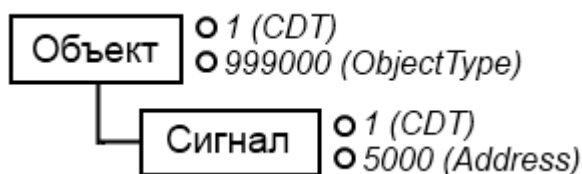
1. Добавьте и настройте [\(стр. 14\)](#) модуль OPC AE Server.
2. Создайте файл конфигурации [\(стр. 30\)](#) сервера <Имя сервера>ExternalObjects.xml в папке сервера C:\Program Files\SePlatform\SePlatform.Server\Server\.
3. Создайте объект [\(стр. 16\)](#) и его сигналы.



ПРИМЕЧАНИЕ

Объектом может являться как отдельный сигнал, так и папка, содержащая несколько сигналов.

Структура с обязательными свойствами приведена на рисунке ниже.



В свойстве **999000 (ObjectType)** следует указать тип объекта. Значение в файле конфигурации [\(стр. 30\)](#) должно совпадать со значением атрибута **Name** для тега **Type**.

4. Создайте и настройте сигнал уведомления, т.е запишите значение в свойство сигнала [\(стр. 16\)](#) **5000 (Address)**.
5. Перезапустите службу **SePlatform.Server**.

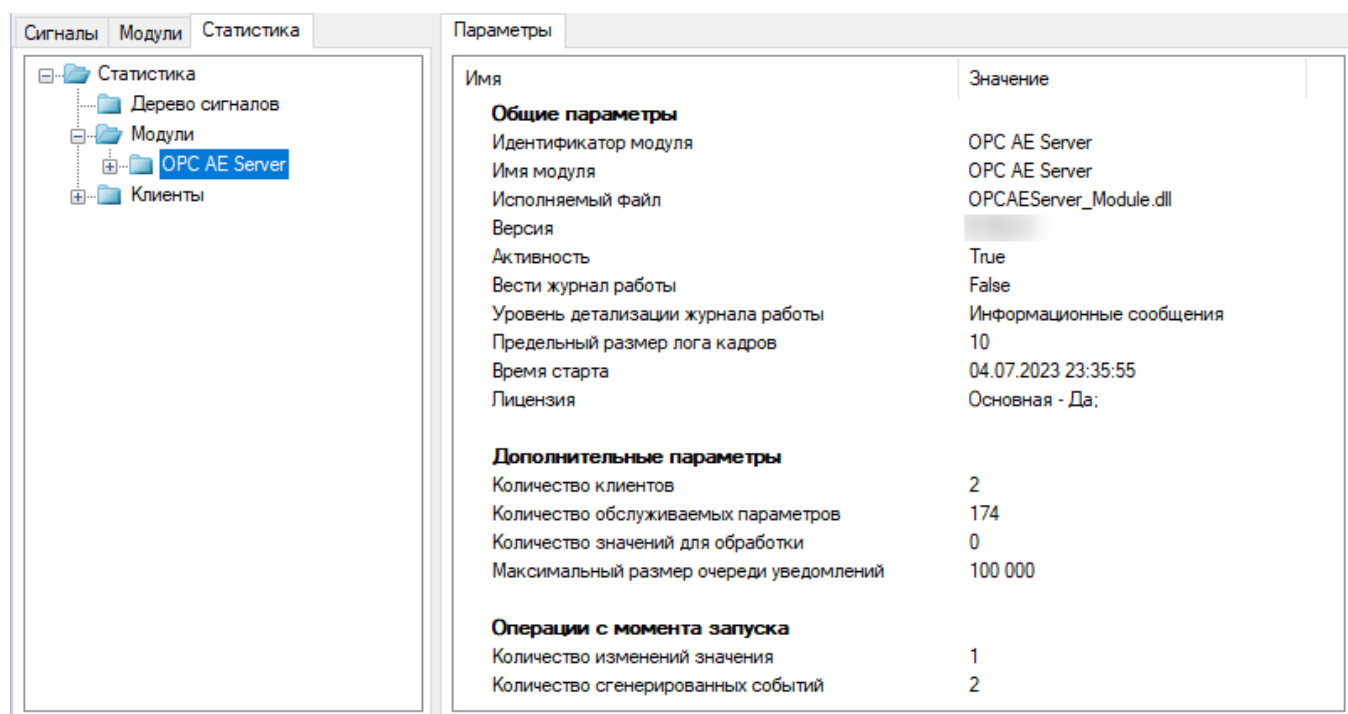
С помощью OPC AE клиента (например, SePlatform.Alarms) можете просматривать сообщения о событиях.

9. Диагностика работы модуля

Статистические данные модуля

Статистическая информация о работе модуля OPC AE Server отображается в сервисном приложении Статистика и на вкладке Статистика сервисного приложения Конфигуратор.

Чтобы просмотреть параметры статистики модуля, подключитесь к SePlatform.Data Server и выберите в дереве модуль OPC AE Server.



Общая статистическая информация модуля представлена в группе **Общие параметры**.

Параметр	Описание
Идентификатор модуля	Идентификатор модуля в конфигурации SePlatform.Data Server.
Имя модуля	Название модуля.
Исполняемый файл	Имя исполняемого файла модуля в каталоге установки SePlatform.Data Server.
Версия	Версия модуля OPC AE Server.
Активность	Активность модуля.
Вести журнал работы	Ведение записи сообщений о работе модуля в журнал работы.
Уровень детализации журнала работы	Типы сообщений, которые фиксируются в журнал приложений.
Предельный размер лога кадров	Размер файла в мегабайтах для записи журнала работы модуля.

Параметр	Описание
Время старта	Время запуска модуля.
Лицензия	Текущее состояние лицензирования модуля.

Статистическая информация модуля о количестве обслуживаемых клиентов и параметров, а также о количестве значений для обработки представлена в группе **Дополнительные параметры**:

Параметр	Описание
Количество клиентов	Количество клиентов, подключенных к модулю.
Количество обслуживаемых параметров	Количество параметров, на которые у модуля есть подписка.
Количество значений для обработки	Количество сообщений, стоящих в очереди для обработки.
Максимальный размер очереди уведомлений	Количество уведомлений, стоящих в очереди на обработку, при достижении, которого приостанавливается работа модуля.

Статистическая информация модуля о количестве операций за время работы модуля представлена в группе **Операции с момента запуска**:

Параметр	Описание
Количество изменений значения	Общее количество изменений значений сигналов за время работы модуля.
Количество сгенерированных событий	Общее количество сгенерированных событий за время работы модуля.

Чтобы просмотреть параметры статистики подключенного клиента модуля OPC AE Server, выберите в дереве интересующий клиент.

The screenshot shows the 'Statistics' tab selected in the top menu. On the left, a tree view shows the hierarchy: 'Статистика' (Statistics) > 'Дерево сигналов' (Signal tree) > 'Модули' (Modules) > 'OPC AE Server' > 'Клиент (ID - 1)' > 'Клиент (ID - 2)'. The 'Клиент (ID - 2)' node is selected. On the right, the 'Parameters' window is open, displaying a table of statistics for the selected client.

Имя	Значение
Общие параметры	
Имя клиента	"DEV70-24-RUB: OpcExplorer (20336)" (ID - 2)
Время подключения	04.07.2023 23:36:00
Количество подписок	0
Количество отправленных событий	0
Количество принятых квитирований	0
Размер очереди для уведомления	0

Статистическая информация подключенного клиента представлена в группе **Общие параметры**.

Параметр	Описание
Имя клиента	Имя подключенного клиента.
Время подключения	Время подключения клиента.
Количество подписок	Количество подписок клиента.
Количество отправленных событий	Количество отправленных событий клиенту.
Количество принятых квитирований	Количество принятых квитирований событий от клиента.
Размер очереди для уведомления	Количество уведомлений в очереди на отправку клиенту.

Журнал работы модуля

Для анализа ошибок, возникающих в процессе работы модуля, воспользуйтесь сервисным приложением Просмотрщик лога кадров.

№	Дата	Время	Описание
13	19.06.2023	20:58:14:635	<-- CModule::OnUninitialize
14	19.06.2023	20:58:14:637	Журнал закрыт
15	19.06.2023	20:58:15:543	Журнал открыт для записи
16	19.06.2023	20:58:15:543	--> CModule::OnInitialize
17	19.06.2023	20:58:15:543	<-- CModule::OnInitialize
18	19.06.2023	20:58:15:584	--> CModule::OnWorkingMode
19	19.06.2023	20:58:15:584	<-- CModule::OnWorkingMode
20	19.06.2023	20:58:15:601	--> CModule::OnStart
21	19.06.2023	20:58:15:609	<-- CModule::OnStart
22	19.06.2023	20:59:03:342	--> CModule::OnWorkingMode
23	19.06.2023	20:59:03:342	<-- CModule::OnWorkingMode
24	19.06.2023	20:59:03:695	--> CModule::OnStop

Список терминов и сокращений

OPC (OLE for process control)	Совокупность спецификаций, определяющих стандартные интерфейсы взаимодействия между программными средствами, используемыми в автоматизации технологических процессов, на основе технологии OLE, COM/DCOM.
OPC AE (OPC Alarms and Events)	Часть OPC, определяющая набор интерфейсов для работы с сообщениями об алармах и событиях.
OPC AE клиент	Программный компонент, заинтересованный в получении уведомлений по интерфейсам AE.
OPC AE сервер	Программный компонент, предоставляющий интерфейс для получения информации о возникающих алармах и событиях.
Аварийное сообщение	Уведомление клиента о некотором (зачастую аварийном) событии в технологическом процессе.
Активная пара серверов	Состояние резервной пары, при котором может выполняться отправка управляющих и регулирующих воздействий.
Квитирование	Подтверждение приёма-передачи уведомления о событии.
Модуль	Программный компонент, работающий в составе сервера ввода/вывода, обеспечивающий некоторую логически законченную функциональность. Основной функцией модулей сервера ввода/вывода является передача данных между компонентами АСУ ТП на уровне SCADA-системы.
Сервер ввода/вывода	Компонент сбора данных и управления технологическим оборудованием в системе автоматизации объектов технологического процесса, предоставляющий доступ к данным другим компонентам системы.
Сигнал	Объект, являющийся носителем информации при обмене данными между компонентами АСУ ТП. Сигнал имеет определенный тип и обладает набором свойств. Основное назначение сигналов - хранить значения реальных физических величин и их свойства: достоверность, параметры доступа и др.
Ядро	Центральная часть сервера ввода/вывода, обеспечивающая связь между модулями и оперативной базой данных.